AMENDMENTS TO THE SPECIFICATION:

Replace paragraph **[0001]** with the following:

[0001] This application claims priority to U.S. Provisional Application Serial No. 60/400,391 titled "JSM Protection," filed Jul. 31, 2002, incorporated herein by reference. This application also claims priority to EPO Application No. 03291907.8, filed Jul. 30, 2003 and entitled "Memory Management Of Local Variables Upon A Change Of Context," incorporated herein by reference. This application also may contain subject matter that may relate to the following commonly assigned co-pending applications incorporated herein by reference: "System And Method To Automatically Stack And Unstack Java Local Variables," Ser. No. 10/632,228[[___]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35422 (1962-05401)~~; "Memory Management Of Local Variables," Ser. No. 10/632,067[[___]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35423 (1962-05402)~~; "A Processor With A Split Stack," Ser. No. 10/632,079[[___]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35425(1962-05404)~~; "Using IMPDEP2 For System Commands Related To Java Accelerator Hardware," Ser. No. 10/632,069[[___]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35426 (1962-05405)~~; "Test With Immediate And Skip Processor Instruction," Ser. No. 10/632,214[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35427 (1962-05406)~~; "Test And Skip Processor Instruction Having At Least One Register Operand," Ser. No. 10/632,084 [[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35248 (1962-05407)~~; "Synchronizing Stack Storage," Ser. No. 10/631,422[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35429 (1962-05408)~~; "Methods And Apparatuses For Managing Memory," Ser. No. 10/631,252[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35430 (1962-05409)~~; "Write Back Policy For Memory," Ser. No. 10/631,185[[___]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35431 (1962-05410)~~; "Methods And Apparatuses For Managing Memory," Ser. No. 10/631,205[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35432 (1962-05411)~~;

"Mixed Stack-Based RISC Processor," Ser. No. 10/631,308[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35433 (1962-05412)~~; "Processor That Accommodates Multiple Instruction Sets And Multiple Decode Modes," Ser. No. 10/631,246[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35434 (1962-05413)~~; "System To Dispatch Several Instructions On Available Hardware Resources," Ser. No. 10/631,585[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35444 (1962-05414)~~; "Micro-Sequence Execution In A Processor," Ser. No. 10/632,216[[____]], filed July 31, 2003~~), Attorney Docket No. TI-35445 (1962-05415)~~; "Program Counter Adjustment Based On The Detection Of An Instruction Prefix," Ser. No. 10/632,222[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35452 (1962-05416)~~; "Reformat Logic To Translate Between A Virtual Address And A Compressed Physical Address," Ser. No. 10/632,215[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35460 (1962-05417)~~; "Synchronization Of Processor States," Ser. No. 10/632,024[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35461 (1962-05418)~~; "Conditional Garbage Based On Monitoring To Improve Real Time Performance," Ser. No. 10/631,195[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35485 (1962-05419)~~; "Inter-Processor Control," Ser. No. 10/631,120[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35486 (1962-05420)~~; "Cache Coherency In A Multi-Processor System," Ser. No. 1-/632,229[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35637 (1962-05421)~~; "Concurrent Task Execution In A Multi-Processor, Single Operating System Environment," Ser. No. 10/632,077[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35638 (1962-05422)~~; and "A Multi-Processor Computing System Having A Java Stack Machine And A RISC-Based Processor," Ser. No. 10/631,939[[____]], filed Jul. 31, 2003~~, Attorney Docket No. TI-35710 (1962-05423)~~.

Replace paragraph **[0029]** with the following:

[0029] The data storage 122 generally comprises data cache ("D-cache") 124 and a data random access memory ("D-RAMset") 126. Reference may be made to copending applications U.S. Ser. No. 09/591,537 filed Jun. 9, 2000 ~~(atty docket TI-29884)~~, Ser. No. 09/591,656 filed Jun. 9, 2000 ~~(atty docket TI-29960)~~, and Ser. No. 09/932,794 filed Aug. 17, 2001 ~~(atty docket TI-31351)~~, all of which are incorporated herein by reference. The stack (excluding the micro-stack 146), arrays and non-critical data may be stored in the D-cache 124, while Java local variables and associated pointers as explained below, as well as critical data and non-Java variables (e.g., C, C++) may be stored in D-RAMset 126. The instruction storage 130 may comprise instruction RAM ("I-RAMset") 132 and instruction cache ("I-cache") 134. The I-RAMset 132 may be used to store "complex" micro-sequenced Bytecodes or micro-sequences or predetermined sequences of code. The I-cache 134 may be used to store other types of Java Bytecode and mixed Java/C-ISA instructions.

Replace paragraph **[0043]** with the following:

[0043] In operation, the processor's core 120~~2~~ may access main memory 106 (FIG. 1) within a given address space. If the information at a requested address in main memory 106 is also stored the data storage 122, the data is retrieved from the data cache 124, 126. If the requested information is not stored in data cache, the data may be retrieved from the main memory 106 and the data cache 124, 126 may be updated with the retrieved data.

Replace paragraph **[0057]** with the following:

[0059] FIG. 9 illustrates one possible embodiment of how to differentiate between local variables associated with finished and unfinished methods. As shown, data array ~~1~~238 generally includes two groups 300 and 302 of local variables. Local variable group 300 generally corresponds to methods that have not yet finished and thus still have significance. Local variable group 302 generally corresponds to methods that have finished and thus do not have any significance.